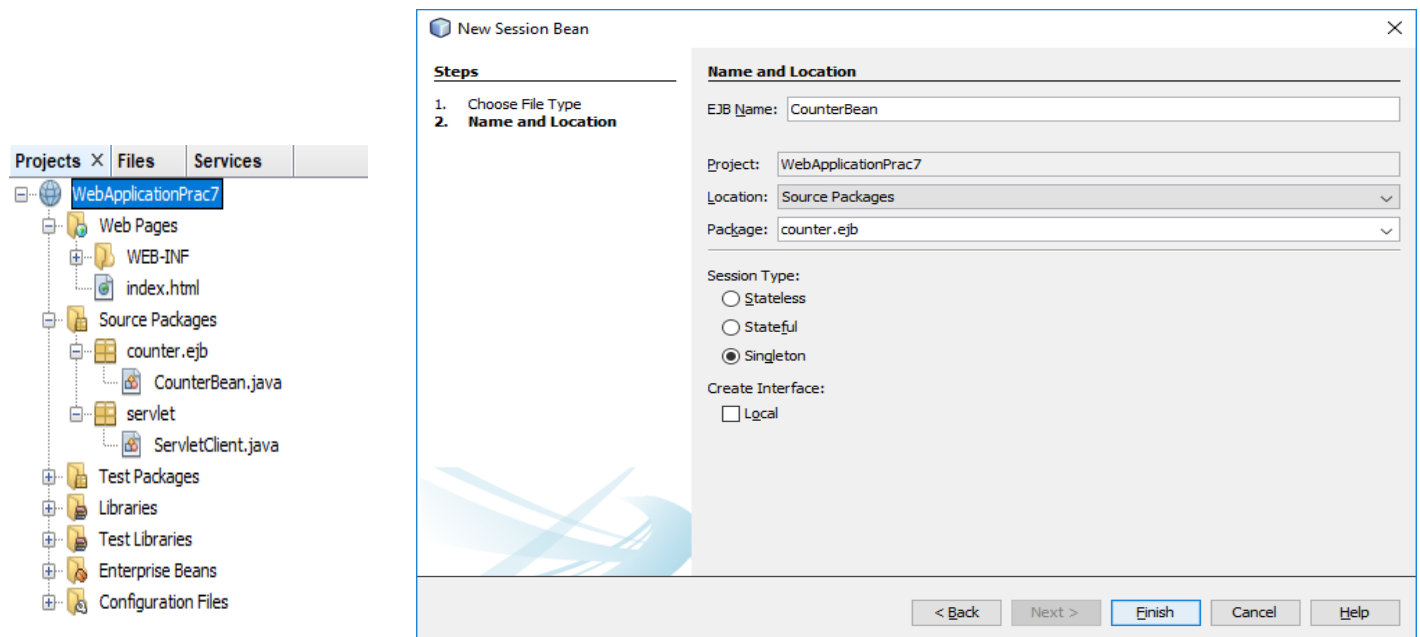


**7a) Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans.****File Name: Index.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Client</title>
    <meta http-equiv="Refresh" content="0; URL=ServletClient">
  </head>
  <body>
  </body>
</html>
```

**File Name: CounterBean.java**

```
package counter.ejb;
import javax.ejb.Singleton;

@Singleton
public class CounterBean {
    private int hits = 1;

    //Increment and return the number of hits
    public synchronized int getHits()
    {
        return hits++;
    }
}
```

**File Name: ServletClient.java**

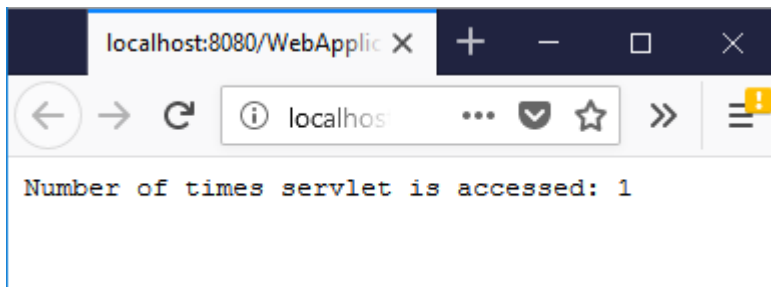
```
package servlet;
```

```
import counter.ejb.CounterBean;
import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

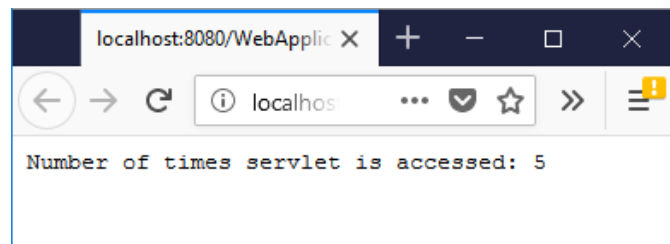
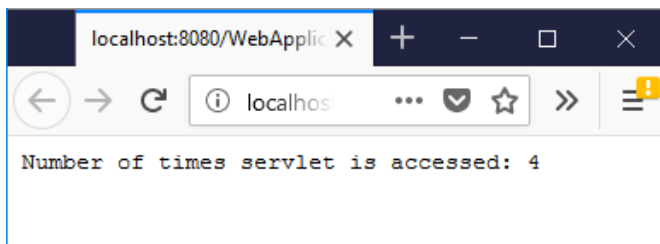
@WebServlet(name = "ServletClient", urlPatterns = {"/ServletClient"})
public class ServletClient extends HttpServlet {

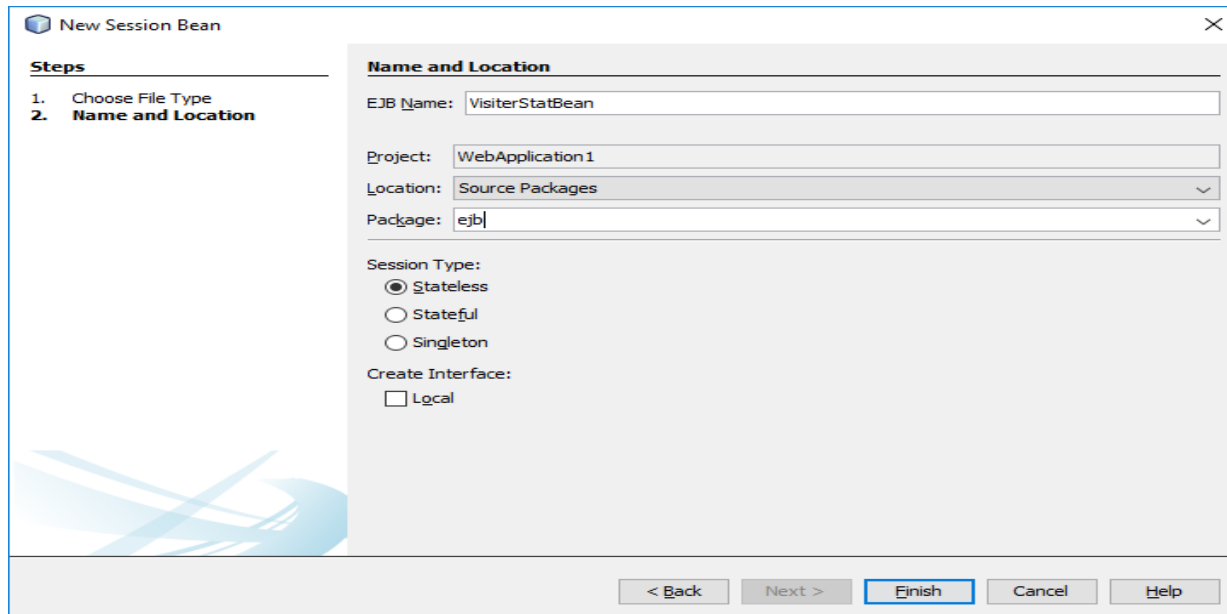
    @EJB CounterBean bean;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out=response.getWriter();
        out.println("Number of times servlet is accessed: " + bean.getHits());
    }
}
```

**Output:**

Refresh the page several times, the counter will increment



**7b) Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean]**

**New Session Bean**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

EJB Name:

Project:

Location:

Package:

Session Type:

☒ Stateless

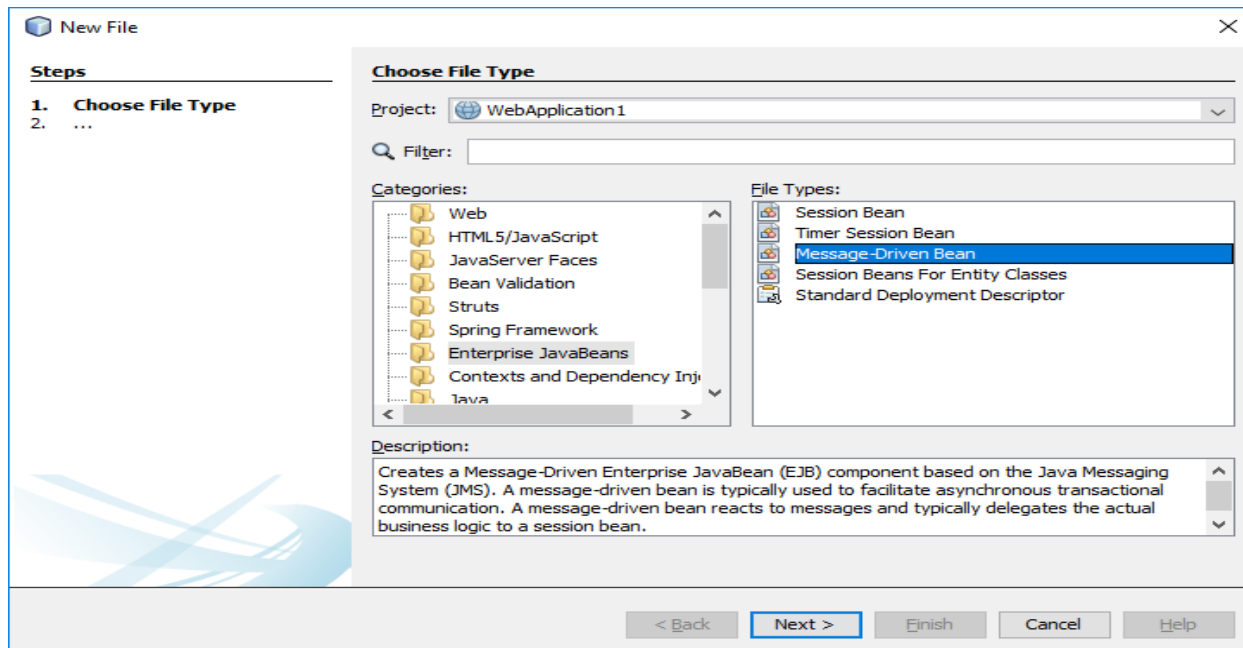
☐ Stateful

☐ Singleton

Create Interface:

☐ Local

< Back Next > **Finish** Cancel Help



**New File**

**Steps**

1. **Choose File Type**
2. ...

**Choose File Type**

Project:

Filter:

Categories:

- Web
- HTML5/JavaScript
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Dependency Injection
- Java

File Types:

- Session Bean
- Timer Session Bean
- Message-Driven Bean**
- Session Beans For Entity Classes
- Standard Deployment Descriptor

Description:

Creates a Message-Driven Enterprise JavaBean (EJB) component based on the Java Messaging System (JMS). A message-driven bean is typically used to facilitate asynchronous transactional communication. A message-driven bean reacts to messages and typically delegates the actual business logic to a session bean.

< Back **Next >** Finish Cancel Help

**New Message-Driven Bean**

**Steps**

1. Choose File Type
2. **Name and Location**
3. Activation Config Properties

**Name and Location**

EJB Name:

Project:

Location:

Package:

☒ Project Destinations:

☐ Server Destinations:

**Valid message destination must be selected.**

**Add Message Destination**

Destination Name:

Destination Type: ☒ Queue ☐ Topic

**New Message-Driven Bean**

**Steps**

1. Choose File Type
2. **Name and Location**
3. Activation Config Properties

**Name and Location**

EJB Name:

Project:

Location:

Package:

☒ Project Destinations:

☐ Server Destinations:

**New File**

**Steps**

1. Choose File Type
2. Name and Location
3. **Activation Config Properties**

**Activation Config Properties**

Property Name	Property Value
acknowledgeMode	AUTO_ACKNOWLEDGE
clientId	
connectionFactoryLookup	
destinationType	QUEUE
destinationLookup	
messageSelector	
subscriptionDurability	NON_DURABLE
subscriptionName	

< Back   Next >   **Finish**   Cancel   Help

**Projects** | **Files** | **Services**

**WebApplication1**

- Web Pages
  - WEB-INF
    - index.jsp
- Source Packages
  - ejb
    - MessageBean.java
    - VisitorStatBean.java
- Test Packages
- Libraries
  - mysql-connector-java-5.1.26-bin.jar
  - JDK 1.8 (Default)
  - GlassFish Server 4.1.1
- Test Libraries
- Enterprise Beans
- Configuration Files
- Server Resources

### MySQL Queries

CREATE DATABASE visitorstat;

USE visitorstat;

```
CREATE TABLE UserStat(
    firstvisitdt timestamp NOT NULL,
    hostname varchar(30) PRIMARY KEY,
    visits int);
```

### File Name: Index.jsp

```
<%@page import="javax.jms.Connection"%>
<%@page import="javax.jms.JMSEException"%>
<%@page import="javax.naming.InitialContext"%>
```

```

<%@page import="javax.jms.MessageProducer"%>
<%@page import="javax.jms.TextMessage"%>
<%@page import="javax.jms.Session"%>
<%@page import="javax.jms.Queue"%>
<%@page import="javax.jms.ConnectionFactory"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
    private static ConnectionFactory connectionFactory;
    private static Queue queue;

    Connection connection = null;
    Session mysession = null;
    MessageProducer messageProducer = null;
    TextMessage message = null;
%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h3>Welcome To Home page</h3>

        <%
            try {
                InitialContext ctx = new InitialContext();
                queue = (Queue) ctx.lookup("jms/Queue2");
                connectionFactory = (ConnectionFactory) ctx.lookup("jms/QueueFactory");

                connection = connectionFactory.createConnection();
                mysession = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
                messageProducer = mysession.createProducer(queue);
                message = mysession.createTextMessage();

                message.setText(request.getRemoteAddr());
                messageProducer.send(message);
            } catch (JMSException e) {
                System.out.println("Exception occurred: " + e.toString());
            }
        %>
    </body>
</html>

```

### File Name: VisitorStatBean.java

```

package ejb;

import javax.ejb.Stateless;
import java.sql.*;
import javax.annotation.*;

@Stateless
public class VisitorStatBean {

```

```
private Connection conn = null;
private ResultSet rs;
private Statement stmt = null;
private String query = null;

@PostConstruct
public void connect() {
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/visitorstat", "root", "root");
        System.out.println("connection successfull.");
    } catch (Exception e) {
        System.err.println("connection failed");
    }
}

@PreDestroy
public void disconnect() {
    try {
        conn.close();
        System.out.println("connection closed");
    } catch (SQLException e) {
        System.err.println("Cannot close connection: " + e.getMessage());
    }
}

public void addVisitor(String host) {
    try {
        stmt = conn.createStatement();
        query = "INSERT INTO UserStat (hostname, visits) VALUES('" + host + "',1)";
        stmt.executeUpdate(query);
    } catch (SQLException e) {
        try {
            stmt = conn.createStatement();
            query = "UPDATE UserStat SET visits = visits + 1 WHERE hostname = '" + host + "'";
            stmt.executeUpdate(query);
        } catch (SQLException ex) {
            System.err.println("Cannot update: " + ex.getMessage());
        }
    }
}
}
```

**File Name: MessageBean.java**

```
package ejb;
```

```
import javax.annotation.Resource;
import javax.ejb.ActivationConfigProperty;
import javax.ejb.EJB;
import javax.ejb.MessageDriven;
import javax.ejb.MessageDrivenContext;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;
```

```

import javax.jms.TextMessage;

@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/Queue2")
    ,@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
})
public class MessageBean implements MessageListener {
    @EJB
    VisitorStatBean vs;

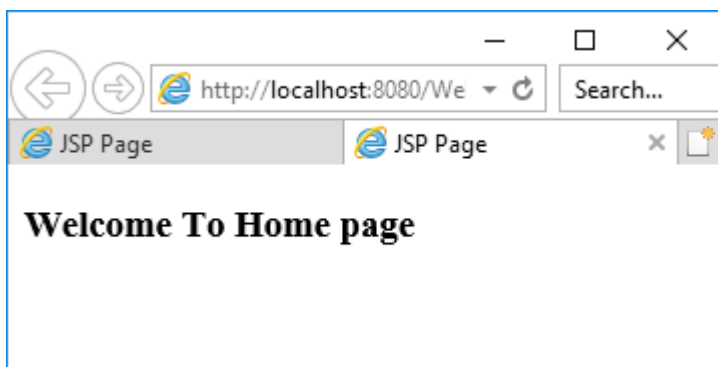
    @Resource
    private MessageDrivenContext mdc;

    public MessageBean() {
    }

    @Override
    public void onMessage(Message message) {
        try {
            if (message instanceof TextMessage) {
                TextMessage msg = (TextMessage) message;
                vs.addVisitor(msg.getText());
            }
        } catch (JMSEException e) {
            mdc.setRollbackOnly();
        }
    }
}

```

### Output



Refresh the webpage several times and the “visits” in the UserStat increases as follows

```
mysql> select * from UserStat;
```

firstvisitdt	hostname	visits
2018-09-03 18:38:57	0:0:0:0:0:0:1	3